

Learning Algorithm for Cross Entropy Function

What is a more simplified way of writing the cross entropy loss function

1. From the previous step, we have $L(\theta) = -[(1 - y)\log(1 - \hat{y}) + y\log(\hat{y})]$
2. Cross entropy loss only makes sense for classification problems
3. The rest of the procedure is the same as the sigmoid neuron, except we use Cross-Entropy to minimize the loss and choose the best parameters w & b
4. **Initialise:** w, b randomly
5. **Iterate over data**
 - a. Compute \hat{y}
 - b. Compute $L(w, b)$ (Where L is the cross-entropy loss function)
 - c. $w_{t+1} = w_t - \eta \Delta w_t$
 - d. $b_{t+1} = b_t + \eta \Delta b_t$
 - e. Pytorch/Tensorflow have functions to compute $\frac{\delta L}{\delta w}$ and $\frac{\delta L}{\delta b}$
6. **Till satisfied**
 - a. Number of epochs is reached (ie 1000 passes/epochs)
 - b. Continue till $\text{Loss} < \epsilon$ (some defined value)
 - c. Continue till $\text{Loss}(w, b)_{t+1} \approx \text{Loss}(w, b)_t$